Fermi National Accelerator Laboratory

FERMILAB-Conf-95/382

Mathematical Modeling of a Fermilab Helium Liquefier Coldbox

M.G. Geynisman and R.J. Walker

Fermi National Accelerator Laboratory

P.O. Box 500, Batavia, Illinois 60510

December 1995

Submitted to the *16th International Cryogenic Engineering Conference*, Kitakyushu, Japan, May 20-24, 1996.

Disclaimer

Mathematical Modeling of a Fermilab Helium Liquefier Coldbox

Michael G. Geynisman and Ronald J. Walker
Fermi National Accelerator Laboratory*
Batavia, Illinois, 60510, USA

ABSTRACT

The Fermilab Central Helium Liquefier (CHL) facility is operated 24 hours-a-day to supply LHe at 4.6°K for the Fermilab Tevatron superconducting proton-antiproton collider Ring and to recover warm return gases. The centerpieces of the CHL are two independent cold boxes rated at 4000 liters/hour and 5400 liters/hour with LN_2 precool. These coldboxes are Claude cycle and have identical heat exchangers trains, but different turbo-expanders. The Tevatron cryogenics demand for higher helium supply from CHL was the driving force to investigate an installation of an expansion engine in place of the Joule-Thompson valve. A mathematical model was developed to describe the thermo- and gas-dynamic processes for the equipment included in the helium coldbox. The model is based on a finite element approach, opposite to a global variables approach, thus providing for higher accuracy and conversion stability. Though the coefficients used in thermo- and gas-dynamic equations are unique for a given coldbox, the general approach, the equations, the methods of computations, and most of the subroutines written in FORTRAN can be readily applied to different coldboxes. The simulation results are compared against actual operating data to demonstrate applicability of the model.

INTRODUCTION

Supporting the world's largest proton/antiproton collider in high energy physics research, the Fermilab Tevatron cryogenic system consists of a hybrid system of a Central Helium Liquefier (CHL) feeding twenty four 1 kW satellite refrigerators through a 6.5 km transfer line and supplying the liquid helium for the superconducting magnets of the accelerator and liquid nitrogen for the thermo shielding [1]. The original CHL system was built in 1979 and consisted of the helium reliquefier plant with one 4000 liters/hour helium liquefier (Coldbox-I) and two reciprocating compressors (Compressor A, B) [2]. Over the years of operations, general upgrades were made to improve the availability of the system, including an addition of a third helium compressor (Compressor C) and 64,000 liters of liquid helium storage [3]. However, a significant impact on the accelerator physics program due to a major CHL failure pointed out the need for the CHL redundancy. Another reason for the system upgrade was the Tevatron accelerator upgrade to 1 TeV energy operations. This is accomplished via lower temperature operation of the magnet system with twenty four cold compressors. The net effect on the cryogenic system is the increase of the CHL load to 170 grams/second, which was beyond the capacity of the Coldbox-I system. Therefore the CHL system was upgraded to include the second cold box (Coldbox-II) and the fourth compressor ("D") [5]. The fourth reciprocating compressor has an increased capacity to 750 grams/second. The Coldbox-II design is generally identical to Coldbox-I. It is tied to the common compressor suction and discharge headers in parallel with Coldbox-I. Both coldboxes have heat exchangers manufactured by ALTEK International, Inc. Coldbox-I has three oil bearing turbo-expanders manufactured by Sulzer Brothers, Ltd. Coldbox-II has three oil bearing turbo-expanders manufactured by Atlas Copco Rotoflow Corp.[4].

Nevertheless, an additional increase in helium production is still desired provided that no major additional capital investment is needed. One of the logical solutions is to replace the Joule-Thompson (JT) valve, which is installed at the coldbox process point of final helium gas expansion,

* Operated by Universities Research Association, Inc. under contract with U.S. Department of Energy

with an expansion engine. This solution is thermodynamically viable, though needs to be evaluated in terms of overall cycle efficiency and production increase versus required capital investment. Some helium liquefier facilities have reported this type of successful conversion in the past [5]. To evaluate this conversion, as well as to have a reliable simulation tool in the future, the authors have developed the mathematical model and implemented it in FORTRAN code. The model describes mathematically the thermodynamic processes in the coldbox equipment and final subcooler at steady-state conditions. The model is a set of non-linear algebraic equations, which describe the links between thermodynamic variables within the coldbox's flow paths. The FORTRAN code of the HEPAK of Cryodata Inc. is used in the model to calculate helium properties. The heat transfer in the coldbox's heat exchangers is computed by using the ALTEC plate-fin geometric data and a customized FEA method. In this method every heat exchanger (HTX) is considered as a set of sections of equal length with a known geometrical data and calculated mass and heat transfer properties. Each section is calculated separately, and then all sections are laced together through "in-out" variables. The non-isentropic gas expansion in the turbo-expanders is computed by using relationships for each turbine's efficiency versus its speed and flow, as per data provided by the turbines' manufacturer. The lacing of the variables along the coldbox's flow paths is accomplished to simulate the propagation of the parameters change due to variation in input variables. Any change in input variables, i.e. coldbox's flow, pressure, temperature, etc., results in coldbox transition to the new and unique set of thermodynamic conditions. The model allows for stable conversion to new conditions at 100% step change of any of the input variables. The overall heat balance of the coldbox is checked to determine the accuracy of conversion. The number of iterations to provide for accurate conversion is less then 10,000.

COLDBOX LAYOUT AND MODEL ASSUMPTIONS

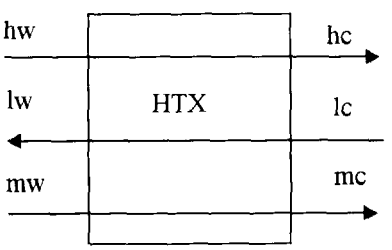
Coldbox-I and -II layouts are identical and shown on Figure 1 further in the text. Both coldboxes have one more 3-path heat exchanger not shown on Figure 1 and located on the warm end. This is a LN_2-to-GHe precool heat exchanger. Though the pressure drops are the most significant for this heat exchanger, its helium gas exit temperature stays at the preset level due to fine regulation via temperature control loop. Therefore we excluded this heat exchanger from the coldbox model, but instead shifted the input variables one heat exchanger down the coldbox.

Notations:

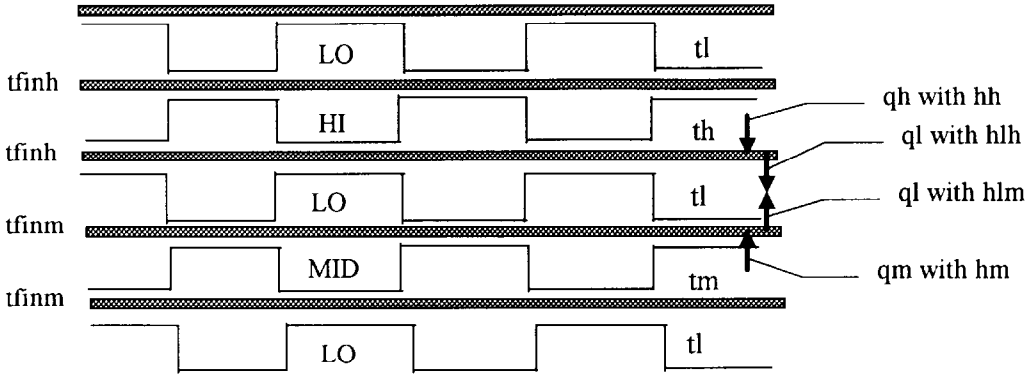
- hw - notation for parameters at the high pressure path and warm end of a heat exchanger, i.e. thw
- hc - notation for parameters at the high pressure path and cold end of a heat exchanger, i.e. thc
- lw - notation for parameters at the low pressure path and warm end of a heat exchanger, i.e. tlw
- lc - notation for parameters at the low pressure path and cold end of a heat exchanger, i.e. tlc
- mw - notation for parameters at the mid pressure path and warm end of a heat exchanger, i.e. tmw
- mc - notation for parameters at the mid pressure path and cold end of a heat exchanger, i.e. tmc

Nomenclature:

- t - temperature, °K
- p - pressure, Pa
- h - enthalpy, J/kg
- s - entropy, J/kg-K
- f - mass flow, kg/s
- ρ - density, kg/m^3



- acfm - actual volumetric flow, ft^3/s or m^3/s
- rpm - rotation speed, revolutions per minute
- μ - viscosity, Pa-s
- c_p - specific heat, J/kg-°K
- q - quality
- qh, qm, ql - heat transferred from high, middle, and to the low pressure paths, watt
- ac - cross section area of flow, m^2
- aw - wetted fin area, m^2
- flen - effective length of the heat exchanger section and the plate fins, m
- de - equivalent diameter, $d_e = \dfrac{4 \cdot ac \cdot flen}{aw}$, m
- qloss - heat load from outside, watt
- tfin - temperatures of a fin, °K
- hh, hm, hlh, hlm - heat transfer coefficients between bulk temperatures in the high path and tfinh, mid path and tfinm, low path and tfinh, and low path and tfinm



- assume that direct heat transfer from mid to high path is negligible. Then the heat transfer equations are:

$$ql = qh + qm$$
$$qh = A \cdot hh \cdot (th - tfinh)$$
$$qm = A \cdot hm \cdot (tm - tfinm)$$
$$ql = \frac{A}{2} \cdot hlh \cdot (tfinh - tl) + \frac{A}{2} \cdot hlm \cdot (tfinm - tl)$$

- the heat transfer coefficients h and the pressure drops Δp for the section of the length dx of the given heat exchanger are calculated with the well-known equations:

$$h = j_t \cdot c_p \cdot G \cdot (\mathrm{Pr})^{-2/3}, \qquad \text{where: } G = f/ac, \text{ mass flow per unit of area}$$
$$\Delta p = f_t \cdot (dx / d_e) \cdot (G^2 / \rho), \qquad \text{where: } \mathrm{Pr} = c_p \mu / k, \text{ k is thermal conductivity}$$

The coefficients j_t and f_t are functions of Re number, where $Re = d_e G / \mu$. The numerical correlation between j_t, f_t and Re are defined per manufacturer test data. In case of the Fermilab coldbox heat exchangers ALTEC provided the authors with the tabulated data for j_t and f_t vs. Reynolds numbers

ranging from 40 to 40,000 for each type of its brazed aluminum plate-fin heat exchangers. Then the authors plotted the tabulated data and fitted it with exponential equations of the following type:

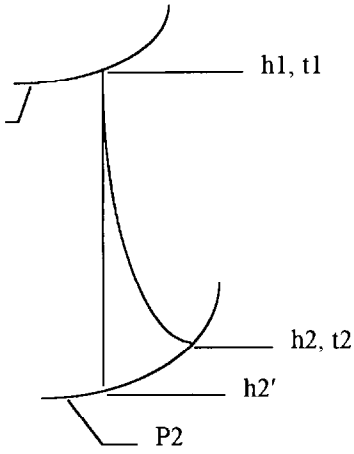
$$j_t = a_0 + a_1 \cdot \exp(-\mathrm{Re}/b_1) + a_2 \cdot \exp(-\mathrm{Re}/b_2) + a_3 \cdot \exp(-\mathrm{Re}/b_3)$$
$$f_t = c_0 + c_1 \cdot \exp(-\mathrm{Re}/d_1) + c_2 \cdot \exp(-\mathrm{Re}/d_2) + c_3 \cdot \exp(-\mathrm{Re}/d_3)$$

- equations describing Atlas Copco Rotoflow Corp. turbo-expander are:

Peripheral blade tip speed at n rpm, m/sec: $U = \dfrac{\pi \cdot n \cdot d_{wheel}}{60}$

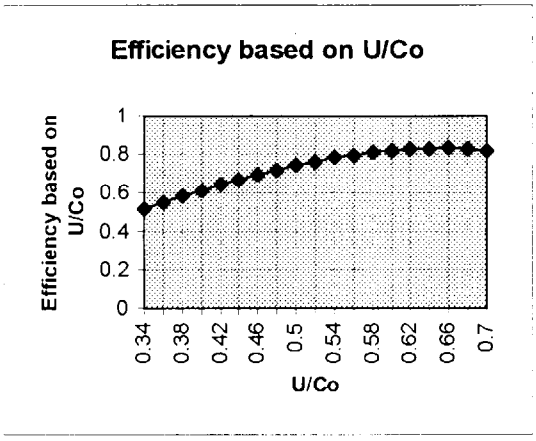
The isentropic efficiency for expander: $\eta_e = \dfrac{h_1 - h_2}{h_1 - h'_2}$

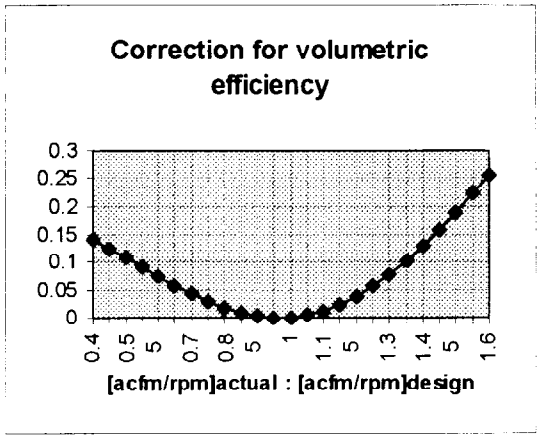
Ideal expansion speed, m/s: $C_o = \sqrt{2 \cdot (h1 - h2')}$



To obtain isentropic efficiency of the turbo-expander the authors have used the set of equations and graphical correlation between efficiency, speed and flow provided by Atlas Copco Rotoflow Corp. The equations are:

$\eta_{tot} = \eta_u - \eta_v$, where η_u is a function of U/Co, and η_v is a function of $(\mathrm{acfm/rpm})^{\mathrm{actual}}/(\mathrm{acfm/rpm})^{\mathrm{design}}$. Then the authors have fitted the graphical dependencies for η_u and η_v for each turbo-expander with 3rd order polynomial equations.





The actual speed of each turbo-expander is a function of its shaft power:

$$n_{actual} = n_{design} \cdot \left[\frac{Power^{\,actual}}{Power^{\,design}}\right]^{1/2.8} = n_{design} \cdot \left[\frac{(f \cdot \Delta h \cdot \eta_{tot})^{actual}}{(f \cdot \Delta h \cdot \eta_{tot})^{design}}\right]^{1/2.8}$$

The graphical representation of the model is shown below on Fig.1 and 2 below.

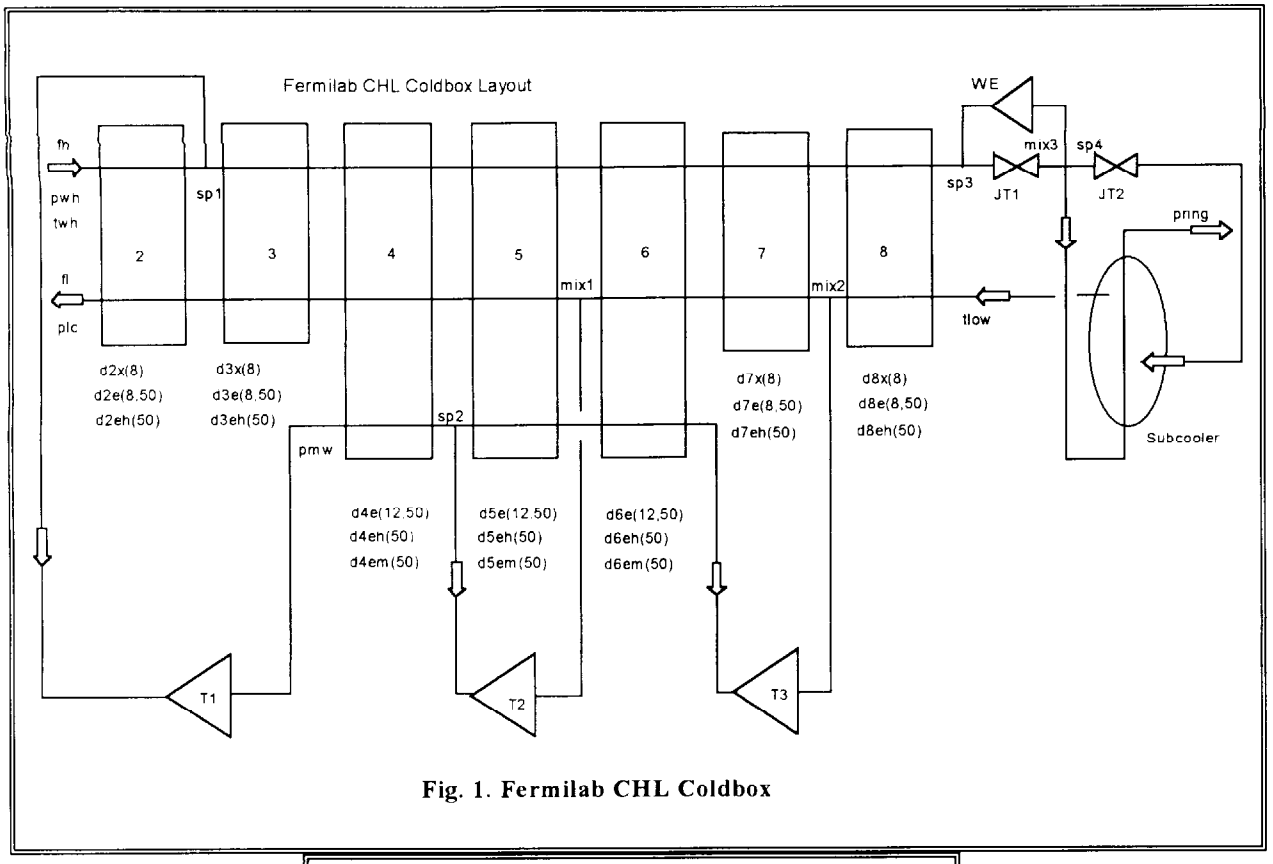


Fig. 1. Fermilab CHL Coldbox

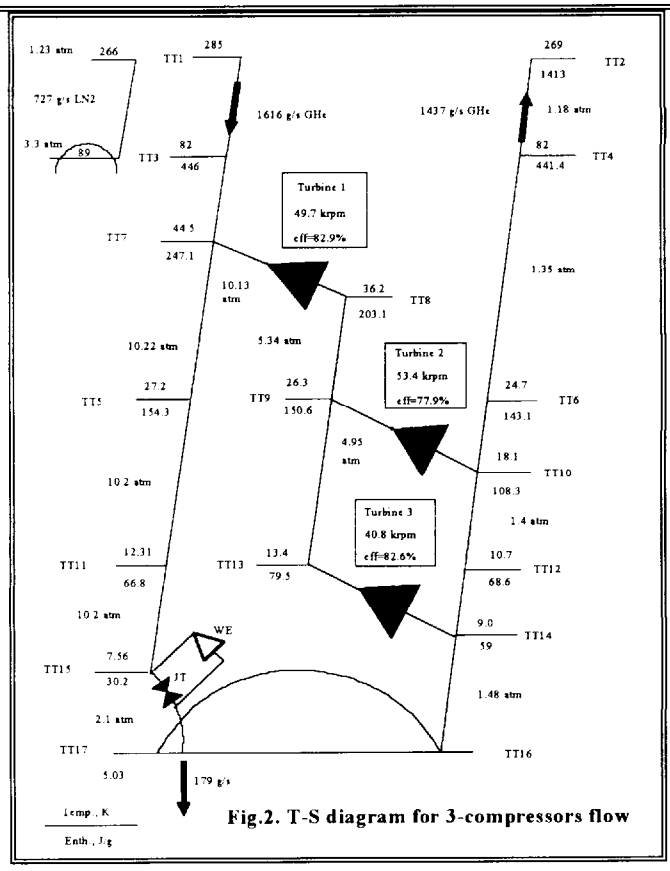
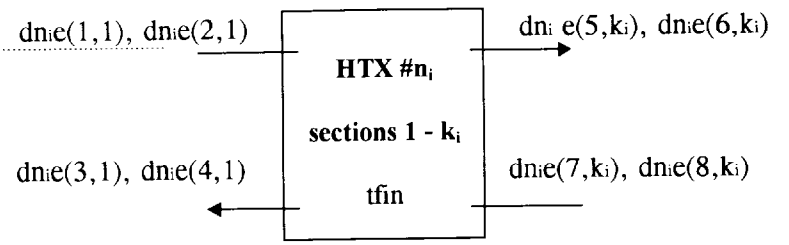


Fig.2. T-S diagram for 3-compressors flow

The model includes 7 heat exchangers, numbered as n_i from i=2 to i=8. The number of cross-sections k_i for each heat exchanger can be set independently. Iteration results have shown that 50 cross-sections is sufficient for most heat exchangers, though the FORTRAN code allows to set that number up to 100.

The following array represents all process parameters for a 2-path heat exchanger, where n_i=2,3,7,8:

phw	thw	plw	tlw	phc	thc	plc	tlc	tfin
$dn_ie(1,1)$	$dn_ie(2,1)$	$dn_ie(3,1)$	$dn_ie(4,1)$	$dn_ie(5,k_i)$	$dn_ie(6,k_i)$	$dn_ie(7,k_i)$	$dn_ie(8,k_i)$	$dn_ieh(k_i)$



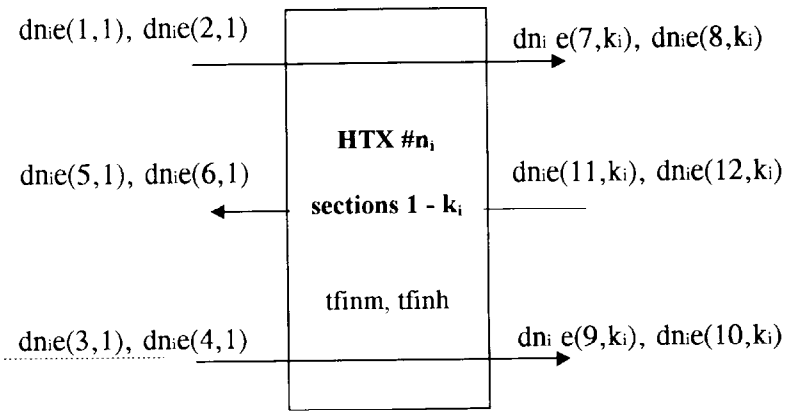
Thc following array represents all geometrical, heat loss and flow parameters for a 2-path heat exchanger, where n_i=2,3,7,8:

awh	awl	ach	acl	flen	qloss	fh	fl
$dn_ix(1)$	$dn_ix(2)$	$dn_ix(3)$	$dn_ix(4)$	$dn_ix(5)$	$dn_ix(6)$	$dn_ix(7)$	$dn_ix(8)$

The following array represents all process parameters for a 3-path heat exchanger, where n_i=4,5,6:

phw	thw	pmw	tmw	plw	tlw
$dn_ie(1,1)$	$dn_ie(2,1)$	$dn_ie(3,1)$	$dn_ie(4,1)$	$dn_ie(5,1)$	$dn_ie(6,1)$

phc	thc	pmc	tmc	plc	tlc	tfinm	tfinh
$dn_ie(7,k_i)$	$dn_ie(8,k_i)$	$dn_ie(9,k_i)$	$dn_ie(10,k_i)$	$dn_ie(11,k_i)$	$dn_ie(12,k_i)$	$dn_iem(k_i)$	$dn_ieh(k_i)$



The following array represents all geometrical, heat loss and flow parameters for a 3-path heat exchanger, where n_i=4,5,6:

awh	awm	awl	ach	acm	acl	flen	qloss	fh
$dn_ix(1)$	$dn_ix(2)$	$dn_ix(3)$	$dn_ix(4)$	$dn_ix(5)$	$dn_ix(6)$	$dn_ix(7)$	$dn_ix(8)$	$dn_ix(9)$

fm	fl
$dn_1x(10)$	$dn_1x(11)$

The following parameters are defined as inputs and can be read from an external file for every run of the model:

- fh = $dn_2x(1)$, high path flow
- sp1 = f1/fh, ratio of turbine 1 flow to the coldbox flow
- sp2 = f2/f1, ratio of turbine 2 flow to turbine 1 flow
- sp3 = fwe/(1-sp1), ratio of wet engine flow to coldbox high side flow
- phw = d2e(1,1), thw = d2e(2,1), parameters of the high path warm end
- pmw = d4e(3,1), midline pressure
- plw = d2e(3,1), pressure of the coldbox low path warm end (compressor suction)
- pring, pressure of the subcooler tube-side flow to the Ring (demand pressure)
- tlow, temperature of the subcooler tube-side flow to the Ring (demand temperature)
- effwe, efficiency of wet engine

STRUCTURE OF THE FORTRAN CODE

The model uses the geometry of each heat exchanger to calculate the heat transfer at each point in the heat exchanger, unlike the programs which use an overall transfer coefficient UA and a log mean temperature difference formulation. This is why it is necessary to go to a relaxation method, because the number of unknowns and equations becomes two times the number of steps or elements, typically 50 to 500. The intention of the authors was to make a program that uses the high pass temperature at the warm end and the low pass temperature at the cold end as inputs. That generates a logical and stable calculation to use the incoming flow parameters as inputs. Experience has shown that using outgoing flow parameters as inputs is prone to experience unphysical inputs that the system can not attain.

Each element is stored and solved, using the above parameters as input parameters. Afterwards, each and all the elements are relaxed by replacing pressure and temperature from above for the high and middle paths, and pressure and temperature from below for the low path. Similarly, the same approach is used for the whole heat exchangers train by propagating calculation results as inputs for next heat exchangers after each step of iteration. For instance, for section $k_i = 32$ of the heat exchanger $n_i = 3$: phw(32) = d3e(1,32) is replaced with phc(31) = d3e(5,31); tlc(32) = d3e(8,32) is replaced with tlw(33) = d3e(4,33); etc.

The model reads in eight data files, one for each heat exchanger with geometrical data and guess values of process parameters for each section, plus a data file for the rest of the input data discussed above. The guess values for the each section process parameters are generated with a separate FORTRAN code based on the known heat exchanger experimental/design global process data, number of sections, and linear distribution. The structure of the input data files is identical to the structure of the output data files of the code. This allows for a sequential iteration when an output data files can be easily copied into an input data files needed later.

The following equations and external subroutines serve to link the process parameters[1]:

[1] Subroutines SHAUG, DFPT, TFDP, etc. are external subroutines from HEPAK FORTRAN code.

- the turbines' input parameters:

```
p11=d2e(5,n2)        - turbine 1 inlet pressure
p21=d4e(3,1)         - turbine 1 exit pressure
t11=d2e(6,n2)        - turbine 1 inlet temperature
p12=d4e(9,n4)        - turbine 2 inlet pressure
p22=d5e(11,n5)       - turbine 2 exit pressure
t12=d4e(10,n4)       - turbine 2 inlet temperature
p13=d6e(9,n6)        - turbine 3 inlet pressure
p23=d7e(7,n7)        - turbine 3 exit pressure
t13=d6e(10,n6)       - turbine 3 inlet temperature
f1=d2x(7)*sp1        - turbine 1 flow
f2=f1*sp2            - turbine 2 flow
f3=f1*(1-sp2)        - turbine 3 flow
```

- turbines' output properties are calculated in special subroutines:

```
CALL TURB1(p11,p21,t11,f1,eff1,t21,dh1,speed1)   - calculates properties for turbine 1
CALL TURB2(p12,p22,t12,f2,eff2,t22,dh2,speed2)   - calculates properties for turbine 2
CALL TURB3(p13,p23,t13,f3,eff3,t23,dh3,speed3)   - calculates properties for turbine 3
```

- heat exchangers' output properties are calculated in special subroutines X2 and X3:

```
CALL X2(d2x,d2e,d2ef,n2,2)                       - calculate properties for HTX2
CALL X2(d3x,d3e,d3ef,n3,3)                       - calculate properties for HTX3
CALL X3(d4x,d4e,d4eh,d4em,n4,4)                  - calculate properties for HTX4
CALL X3(d5x,d5e,d5eh,d5em,n5,5)                  - calculate properties for HTX5
CALL X3(d6x,d6e,d6eh,d6em,n6,6)                  - calculate properties for HTX6
CALL X2(d7x,d7e,d7ef,n7,7)                       - calculate properties for HTX7
CALL X2(d8x,d8e,d8ef,n8,8)                       - calculate properties for HTX8
```

- return flow fl is calculated by propagating from HTX #8 through JT and WE, combining these two flows in the MIX3, and finally calculating the split 4 and return flow fl. First we start with wet engine and JT flows:

```
fwe=fh*(1.-sp1)*sp3                              - wet engine flow
fjt=fh*(1.-sp1)*(1.-sp3)                         - JT flow
```

- then we calculate the JT and wet engine exit conditions:

```
CALL JT(d8e(5,n8),d8e(6,n8),pring,tjt,q)         - isenthalpic expansion in JT valve for
                                                   given inlet conditions, and exit
                                                   pressure
CALL ENG(d8e(5,n8),d8e(6,n8),pring,twe,hwe,effwe) - expansion with given efficiency,
                                                   given inlet conditions, and exit
                                                   pressure
```

- subcooler inlet parameters are calculated in special subroutine MIX. This subroutine reads the parameters of two incoming flows, and calculates the enthalpy and temperature of the mix at a given pressure:

```
CALL MIX(pring,twe,fwe,tjt,fjt,tsub,f)        - calculate subcooler inlet properties
```

- then we calculate the split between flow to the Ring and subcooler flow, which is necessary to maintain the coldbox return temperature tlow at a given level. Here we define flow 'f' as a total flow from HTX8=d8x(7), flow 'fl' as a coldbox return flow, and calculate sp4=fl/f:

```
CALL SUBC(tlow,pring,tsub,f,d8e(7,n8),sp4)    - calculate the subcooler shell side flow
```

- then the Ring and coldbox return flows:

```
fsc= fh*(1.-sp1)*sp4                  - flow to the shell side of the subcooler
fring= fh*(1.-sp1)*(1.-sp4)           - flow to the Ring via the tube side of
                                        the subcooler
fl=fsc                                - return flow to the low path of the
                                        coldbox
```

- then the heat exchangers low path flows:

```
d8x(8)=fl                             - flow through HTX8 low path
d7x(8)=fl+f3                          - flow through HTX7 low path
d6x(11)=fl+f3                         - flow through HTX6 low path
d5x(11)=fl+f3+f2                      - flow through HTX5 low path
d4x(11)=fl+f3+f2                      - flow through HTX4 low path
d3x(8)= fl+f3+f2                      - flow through HTX3 low path
d2x(8)= fl+f3+f2                      - flow through HTX2 low path
```

- heat exchangers middle path flows:

```
d4x(10)=f1                            - flow through HTX4 mid path
d5x(10)=f3                            - flow through HTX5 mid path
d6x(10)=f3                            - flow through HTX6 mid path
```

- then we update for the first time the input parameters:

```
d2e(1,1)=phw                          - HTX2 high path inlet pressure
d2e(2,1)=thw                          - HTX2 high path inlet temperature
d4e(3,1)=pmw                          - HTX4 mid path inlet pressure
d8e(8,n8)=tlow                        - HTX8 low path inlet temperature
d2e(3,1)=plw                          - HTX2 low path exit pressure
d2x(7)=fh                             - HTX2 high path flow
d3x(7)=fh*(1-sp1)                     - HTX2 high path flow
d4x(9)= fh*(1-sp1)                    - HTX2 high path flow
d5x(9)= fh*(1-sp1)                    - HTX2 high path flow
d6x(9)= fh*(1-sp1)                    - HTX2 high path flow
d7x(7)= fh*(1-sp1)                    - HTX2 high path flow
```

d8x(7)= fh*(1-sp1) - HTX2 high path flow
d4e(4,1)=t21 - HTX4 mid path warm end = turbine 1 exit temperature

- HTX5 and HTX7 low path cold end temperatures are calculated with a special subroutine MIX. This subroutine reads the parameters of two incoming flows, and calculates the enthalpy and temperature of the mix at a given pressure:

CALL MIX(d5e(11,n5),t22,f2,d6e(6,1),d6x(11),d5e(12,n5),d5x(11)) - for HTX5
CALL MIX(d7e(7,n7),t23,f3,d8e(4,1),d8x(8),d7e(8,n7),d7x(8)) - for HTX7

Additionally, we can now complete lacing for the heat exchangers pressures and temperatures. This lacing propagates from heat exchanger #2 to heat exchanger #8 in the high path, from heat exchanger #4 to heat exchanger #6 in the midlle path, and from heat exchanger #8 to heat exchanger #2 in the low path. For low path the compressor suction pressure plw is fixed, thus lacing for pressures propagates from heat exchanger #2 to heat exchanger #8, opposite the direction of the helium flow.

from HTX #2 to #3:	from HTX #3 to #4:
d3e(1,1)=d2e(5,n2)	d4e(1,1)=d3e(5,n3)
d3e(2,1)=d2e(6,n2)	d4e(2,1)=d3e(6,n3)
d3e(3,1)=d2e(7,n2)	d4e(5,1)=d3e(7,n3)
d2e(8,n2)=d3e(4,1)	d3e(8,n3)=d4e(6,1)

from HTX #4 to #5:	from HTX #5 to #6:
d5e(1,1)=d4e(7,n4)	d6e(1,1)=d5e(7,n5)
d5e(2,1)=d4e(8,n4)	d6e(2,1)=d5e(8,n5)
d5e(5,1)=d4e(11,n4)	d6e(5,1)=d5e(11,n5)
d4e(12,n4)=d5e(6,1)	d5e(12,n5)=already defined in MIX1
d5e(3,1)=d4e(9,n4)	d6e(3,1)=d5e(9,n5)
d5e(4,1)=d4e(10,n4)	d6e(4,1)=d5e(10,n5)

from HTX #6 to #7:	from HTX #7 to #8:
d7e(1,1)=d6e(7,n6)	d8e(1,1)=d7e(5,n7)
d7e(2,1)=d6e(8,n6)	d8e(2,1)=d7e(6,n7)
d7e(3,1)=d6e(11,n6)	d8e(3,1)=d7e(7,n7)
d6e(12,n6)=d7e(4,1)	d7e(8,n7)=already defined in MIX2

By this time all lacing between the heat exchangers is completed. Then we make series of checks or counting to decide whether the coldbox equations have already converged or not. Additionally we calculate the heat balance for the complete coldbox. Ballance is:

$$B = fh \cdot hhw - fring \cdot hring - fl \cdot hlw + \sum_{i=2}^{8} radiation_i - \sum_{i=1}^{3} f_i \cdot \Delta hturbine_i$$

where: hhw=coldbox HTX2 high path inlet enthalpy; hring=enthalpy of the ring flow; hlw=entalpy of the return flow; Δhturbine$_i$=enthalpy drop in turbine$_i$.

DESCRIPTION OF SUBROUTINES FOR COLDBOX ELEMENTS

A) Subroutines X2 and X3. These subroutines are for 2-path and 3-path heat exchangers respectively. They use an internal loop to find the solution for parameters at the low path warm end and high path

cold end for each section of the heat exchanger. Subroutine X3 additionally finds the solution for parameters at the middle path cold end for each section of the 3-path heat exchanger. These solutions are used to update array of parameters dme. Each 2-path or 3-path heat exchanger has a unique geometrical data and unique number of sections, thus the subroutine X2 or X3 is called respectively for each heat exchanger with unique array of geometrical data dmx. Note that the defining inputs at the warm end of the high path and the middle path do not need to be the same. Furthermore, the cold end parameters do not in general come out to be the same for the high path and the middle path. The authors do not know of any other calculations of 3-path heat exchanger which provides for this generality. To calculate temperatures of the fins X2 and X3 call in the external subroutine HEAT in each step of iteration. The heat transfer for each element is calculated via enthalpy change. External heat loss qloss is assumed as distributed equally through all sections of a given heat exchanger. After new process parameters have been calculated for each section, the sections are laced together so that the input corners influence can propagate through the heat exchanger toward the output end. Below is an example of subroutine X3:

```
SUBROUTINE x3(d4x,d4e,d4eh,d4em,n,kkk)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION d4e(12,n),d4x(11),d4eh(n),d4em(n)
```

here: n - number of sections; kkk - an index indicating the type of the heat exchanger to select the right curves for heat transfer and Δp coefficients.

```
fh=d4x(9)      - defining the high path flow
fm=d4x(10)     - defining the mid path flow
fl=d4x(11)     - defining the low path flow
```

We begin the element loop, find a solution for lw and hc, mc variables for each element and stuff it in the array dme to update this array. First the geometry calculations:

```
awh=d4x(1)          - wetted fin area for high path
awm=d4x(2)          - wetted fin area for mid path
awl=d4x(3)          - wetted fin area for low path
ach=d4x(4)          - flow cross section area for high path
acm=d4x(5)          - flow cross section area for mid path
acl=d4x(6)          - flow cross section area for low path
flen=d4x(7)         - effective length of the heat exchanger
deh=4.*(ach*flen/awh)  - hydraulic diameter for the high path
dem=4.*(acm*flen/awm)  - hydraulic diameter for the mid path
del=4.*(acl*flen/awl)  - hydraulic diameter for the low path
step=n              - number of sections in the heat exchanger
dx=flen/step        - length of one section
qloss=d4x(8)        - radiation loss
```

Then each element is solved separately within an outer loop:

```
DO 99 i=1,n

phw=d4e(1,i)    tmw=d4e(4,i)    phc=d4e(7,i)
thw=d4e(2,i)    plw=d4e(5,i)    pmc=d4e(9,i)
pmw=d4e(3,i)    tlw=d4e(6,i)    thc=d4e(8,i)
```

```
tmc=d4e(10,i)        tlc=d4e(12,i)        tfinm=d4em(i)
plc=d4e(11,i)        tfinh=d4eh(i)
```

Then we begin the element loop, converge each element based on stability in tlw, thc, tmc. Later we decided to remove the loop to speed up calculations without decreasing the stability.

```
      ielem=0
370   continue
      ielem=ielem+1
      thcsav=thc
      tmcsav=tmc
      tlwsav=tlw
```

At this point we begin the inner loop, fin loop. Its function is to iterate tfinx until they converge.

```
      ifin=0
386   continue

      ifin=ifin+1          th=(thw+thc)/2.      pm=(pmw+pmc)/2.
      savh=tfinh           ph=(phw+phc)/2.      tl=(tlw+tlc)/2.
      savm=tfinm           tm=(tmw+tmc)/2.      pl=(plw+plc)/2.
```

We use indexes for: index=1 for high path, =2 for low path, and =3 for middle path:

```
      index=1
      CALL HEAT(fh,tfinh,th,ph,ach,deh,hh,dpdxh,kkk,index)
      index=3
      CALL HEAT(fm,tfinm,tm,pm,acm,dem,hm,dpdxm,kkk,index)
      index=2
      CALL HEAT(fl,tfinh,tl,pl,acl,del,hlh,dpdxl,kkk,index)
      tfinh=(awh*th*hh+awl/2.*tl*hlh)/(awh*hh+awl/2.*hlh)
      index=2
      CALL HEAT(fl,tfinm,tl,pl,acl,del,hlm,dpdxl,kkk,index)
      tfinm=(awm*tm*hm+awl/2.*tl*hlm)/(awm*hm+awl/2.*hlm)

      if(ifin.gt.100) goto 4000
      if(abs(savh-tfinh).gt.1.e-4*tfinh)go to 386
      if(abs(savm-tfinm).gt.1.e-4*tfinm)go to 386
```

This is the end of the fin loop.

```
4000  continue

      dpl=dpdxl*dx         qh=awh*hh*(th-tfinh)/step
      dph=dpdxh*dx         qm=awm*hm*(tm-tfinm)/step
      dpm=dpdxm*dx
```

Then we calculate the heat transferred into each element and update the element. Subroutine DFPT finds density from given pressure and temperature. Subroutine SHAUG finds entropy and enthalpy

from given density, pressure, and temperature. Subroutine DTFPX finds density and temperature from given pressure and enthalpy.

```
CALL DFPT(ID,dhw,Q,phw,thw)
CALL SHAUG(S,ehw,A,U,G,phw,dhw,thw)
CALL DFPT(ID,dmw,Q,pmw,tmw)
CALL SHAUG(S,emw,A,U,G,pmw,dmw,tmw)
phc=phw-dph
ehc=ehw-qh/fh
CALL DTFPX(ID,dhc,thc,Q,DL,DV,DPDDDTS,phc,ehc,'PH')
pmc=pmw-dpm
emc=emw-qm/fm
CALL DTFPX(ID,dmc,tmc,Q,DL,DV,DPDDDTS,pmc,emc,'PH')
plc=plw+dpl
CALL DFPT(ID,dlc,Q,plc,tlc)
CALL SHAUG(S,elc,A,U,G,plc,dlc,tlc)
elw=elc+(qh+qm+qloss/step)/fl
CALL DTFPX(ID,dlw,tlw,Q,DL,DV,DPDDDTS,plw,elw,'PH')
```

By this point the output parameters have been computed from the input corners. The next step is to load the results into the process array $d4e_{ie}$ and fin arrays $d4em_{iem}$ and $d4eh_{ieh}$.

```
if(ielem.gt.100) goto 5000
if(abs(thcsav-thc).gt.1.e-4*thc)go to 370
if(abs(tmcsav-tmc).gt.1.e-4*tmc)go to 370
if(abs(tlwsav-tlw).gt.1.e-4*tlw)go to 370

5000 continue

d4e(11,i)=plc        d4e(9,i)=pmc        d4eh(i)=tfinh
d4e(6,i)=tlw         d4e(8,i)=thc        d4em(i)=tfinm
d4e(7,i)=phc         d4e(10,i)=tmc

99   continue

     DO 197 i=1,n-1

     d4e(5,i+1)=d4e(11,i)          d4e(3,i+1)=d4e(9,i)
     d4e(12,i)=d4e(6,i+1)          d4e(2,i+1)=d4e(8,i)
     d4e(1,i+1)=d4e(7,i)           d4e(4,i+1)=d4e(10,i)

197  continue
     RETURN
     END
```

B) Subroutine HEAT. This subroutine calculates heat transfer coefficient hh and pressure drop coefficient dpdx for a given set of process parameters and geometrical data of a single section of a plate-fin heat exchanger. The subroutine uses the same equations and coefficients for every section of a given heat exchanger, but unique equations and coefficients for each of the coldbox heat exchanger. This subroutine needs Reynolds and Prandtl numbers, which are calculated based on mass flow and

mean temperature between fins and bulk fluid. Density, viscosity, thermal conductivity, and heat capacity at constant pressure calculations are done using HEPAK external subroutines.

```
SUBROUTINE HEAT(fh,tfin,th,ph,ach,deh,hh,dpdx,kkk,index)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

fmdot=fh                - note that the label 'h' for the high pass is used in the argument list,
                          though the calling argument list will reflect the correct pass.

g=fmdot/ach             - mass flow rate per unit area
tempm=0.5*(tfin+th)     - 'm' is used for mean temperature between wall and bulk fluid which
                          should be used to calculate properties.

CALL DFPT(id,densm,q,ph,tempm)   - gives the density at the mean temperature
CALL DFPT(id,densb,q,ph,th)      - gives the density at the bulk temperature

fmu=viscos(densm,tempm)          - absolute viscosity at mean temperature, Pa/s
fmub=viscos(densb,th)            - absolute viscosity at bulk temperature, Pa/s
Re=deh*g/fmu                     - Reynolds number
cpp=cp(densm,tempm)              - heat capacity at mean temperature
tcond=tcon(densm,tempm)          - thermal conductivity of gas at mean temperature
Pr=cpp*fmu/tcond                 - Prandlt number
```

The following logical statements define the selection of the specific curve for heat transfer and Δp coefficients:

```
      if(kkk.eq.2.and.index.eq.1) goto 100   - HTX2 high path
      if(kkk.eq.2.and.index.eq.2) goto 101   - HTX2 low path
      if(kkk.eq.3.and.index.eq.1) goto 100   - HTX3 high path
      ..... etc.....
      if(kkk.eq.7.and.index.eq.2) goto 101   - HTX7 low path
      if(kkk.eq.7.and.index.eq.1) goto 102   - HTX8 high path
      if(kkk.eq.6.and.index.eq.3) goto 100   - HTX8 low path

100   dpdx=Function(Re, densm, deh, g)[2]
      hh=Function(Re, Pr, cpp, g)[2]
      goto 103
101   dpdx=Function(Re, densm, deh, g)[2]
      hh=Function(Re, Pr, cpp, g)[2]
      goto 103
102   dpdx=Function(Re, densm, deh, g)[2]
      hh=Function(Re, Pr, cpp, g)[2]
      goto 103
103   continue

      RETURN
      END
```

[2] The coefficients of the equations are ALTEC International proprietary data

C) Subroutines TURB1,2,3. These subroutines calculate efficiency of the respective expander as dH/dH_{ideal} based on the data provided by Rotoflow for U/C and volumetric efficiencies. This subroutine calculate expander exit temperature based on inlet pressure, temperature, flow, and exit pressure. Below is an example of subroutine TURB1.

```
SUBROUTINE TURB1(p1,p2,t1,fl,eff,t2,dh,speed)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
```

Notations and definitions: '1' = inlet, '2' = exit parameters, 'd' = design, 'a' = actual. Input parameters are: p1, p2, t1, fl. Output parameters are: eff, t2, dh=h1-h2, speed. Additionally: uc = ratio of tip speed U over theoretical speed C0; dspeed=design speed; dvol=design volume flow; duc=design U/C0; deff=design efficiency. At fist, we read eff and speed as iteration guess values. Then we define the design values:

```
dspeed=55000.          duc=0.66          dflow=1.2820
dvol=0.1651847         deff=0.82         ddh=55451.274
```

Then we compute the ideal expander, ds=0:

```
      j=0

      CALL DFPT(id,d1,q,p1,t1)                          - inlet density
      CALL SHAUG(s1,h1,a,u,g,p1,d1,t1)                  - inlet entropy
      s2=s1                                             - exit entropy
      CALL DTFPX(id,d2,t2,q,dl,dv,dpdts,p2,s2,'PS')     - exit density and temperature
      CALL SHAUG(s2,h2i,a,u,g,p2,d2,t2)                 - exit enthalpy h2i at ideal expansion
      dhi=h1-h2i                                        - ideal enthalpy change
      ac=223.837*SQRT(dhi*4.29927e-4)                   - actual theoretical speed C

6     j=j+1
      aspeed=dspeed*((fl*dhi*eff)/(dflow*ddh*deff))**(1./2.8)   - actual rpm
      au=0.0130899*aspeed                               - actual tip speed
      auc=au/ac                                         - actual U/C
      effu=Function(auc)[3]                             - actual effu

      h2=h1-effu*(h1-h2i)                               - actual exit enthalpy h2
      CALL DTFPX (id,d2,t2,q,dl,dv,dpdts,p2,h2,'PH')    - actual exit density and temperature

      avol=fl/d2                                        - volumetric flow, m3/sec
      xv=(avol*aspeed)/(dvol*dspeed)                    - actual to design speed and flow ratio
      deffv=Function(xv)[3]                             - drop in efficiency due to flow
      efft=(effu-deffv)                                 - 'total' efficiency from effu and deffv

      aspeed=dspeed*((fl*dhi*efft)/(dflow*ddh*deff))**(1./2.8)   - next iteration of the
                                                                   actual speed

      if(abs(efft-eff).le.1.e-5)go to 5                 - then we check for convergence and go
      eff=efft                                            back to the beginning of the loop if
```

[3] Atlas Copco Rotoflow Corporation proprietary data

```
      if(j.gt.20)go to 5                    the convergence condition is not
      goto 6                                satisfied
5     eff=efft
      dh=h1-h2
      speed=aspeed
      RETURN
      END
```

D) Subroutine SUBC. This subroutine adjusts the flow to the subcooler to maintain a given temperature (i.e. 4.7°K) of the two phase helium routed back to the heat exchanger #8 on the low path, tlow. It is assumed that the subcooler is perfect and sends liquid helium to the Ring at the temperature tlow. Below is example of subroutine SUBC.

```
      SUBROUTINE SUBC(tlow,pring,tsub,f,plow,sp4)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
```

Notations are: pring = pressure to the Ring, input data (assumption is that incoming pressure is pring, i.e. no pressure drop); tsub = temperature coming in to the subcooler, input data; f = cold flow coming out of the JT and WE, same as high side of heat exchanger #8, same as coming into the subcooler, input data; flow = flow returning to HTX8, output data; plow = pressure at the low pressure cold end of heat exchanger #8, calculated data (from the total pressure drop in the coldbox low side); split4 = fractional flow to the subcooler, comes out as actual value. We assume that the throttling on the subcooler JT is isenthalpic from pring to plow.

```
      tsatv=TSATFP(plow)                    - saturation temperaure at plow
      if (tlow.lt.tsatv) print 559          - check for liquid fraction
559   FORMAT(' Warning!!! 2 phase in HTX8!!!')   - warning message

      CALL DFPT(id,dsub,q,pring,tsub)       - density of the tube side flow
      CALL SHAUG(ssub,hsub,a,u,g,pring,dsub,tsub)   - enthalpy of the tube side flow
      CALL DFPT(id,dlow,q,plow,tlow)        - density of the shell side flow
      CALL SHAUG(s,hlow,a,u,g,plow,dlow,tlow)   - enthalpy of the shell side flow
      CALL DFPT(id,dring,q,pring,tlow)      - density of the flow to the Ring
      CALL SHAUG(s,hring,a,u,g,pring,dring,tlow)   - enthalpy of the flow to the Ring
```

Then the flow balance and heat balance give the solution for fl to the shell side of the subcooler:

```
      fl=f*(hsub-hring)/(hlow-hring)
      sp4=fl/f

      RETURN
      END
```

E) Subroutines JT, MIX, and ENG calculate the exit properties respectively for: a) isenthalpic expansion in JT valve; b) mix of two mass flows with known parameters; c) expansion in expansion engine of known isentropic efficiency. These subroutines are straight-forward and use HEPAK external subroutines to calculate properties.

RESULTS OF THE SIMULATION

As it was said earlier, the structure of the output data files generated by the model is identical to the structure of the input data files read into the code. The model generates eight data files, one for each heat exchanger with geometrical data and final values of process parameters for each section, plus a data file for the coldbox global input data. The typical coldbox global output data file generated by the code is shown below (all parameters in SI units).

COLDBOX INPUT:

sp1	sp2	sp3	sp4	effwe	imain	fh	thw	phw	pmw
0.7746	1.0000	0.7500	0.5000	0.3445	5000	1.400	82.000	0.13100E+07	0.62050E+06

pring	tlow	plw
0.3309E+06	0.1217E+06	4.7

COLDBOX OUTPUT:

turb1 speed	turb2 speed	turb3 speed	turb1 eff	turb2 eff	turb3 eff
52429.75	49218.15	40276.52	0.7601	0.7314	0.7547

t11, turb1 in	t21, turb1 out	t12, turb2 in	t22, turb2 out	t13, turb3 in	t23, turb3 out
45.26	35.79	25.63	16.33	12.87	8.00

f1	f2	f3	fwe	fjt	fring
1.08437	0.54219	0.54219	0.31563	0.00000	0.20688

HEAT EXCHANGERS OUTPUT DATA:

HTX# HTX#	phw phc	thw thc	plw plc	tlw tlc	pmw pmc	tmw tmc
2	0.1900E+03	0.8200E+02	0.1766E+02	0.8198E+02		
2	0.1890E+03	0.4526E+02	0.1900E+02	0.3835E+02		
3	0.1890E+03	0.4526E+02	0.1900E+02	0.3835E+02		
3	0.1890E+03	0.3762E+02	0.1909E+02	0.3625E+02		
4	0.1890E+03	0.3762E+02	0.1909E+02	0.3625E+02	0.9000E+02	0.3579E+02
4	0.1876E+03	0.2363E+02	0.1971E+02	0.2291E+02	0.8894E+02	0.2563E+02
5	0.1876E+03	0.2363E+02	0.1971E+02	0.2291E+02	0.8894E+02	0.2563E+02
5	0.1875E+03	0.1634E+02	0.2013E+02	0.1633E+02	0.8854E+02	0.1634E+02
6	0.1875E+03	0.1634E+02	0.2013E+02	0.1634E+02	0.8854E+02	0.1634E+02
6	0.1875E+03	0.1292E+02	0.2022E+02	0.1115E+02	0.8817E+02	0.1287E+02
7	0.1875E+03	0.1292E+02	0.2022E+02	0.1115E+02		
7	0.1873E+03	0.8002E+01	0.2056E+02	0.8002E+01		
8	0.1873E+03	0.8002E+01	0.2056E+02	0.8011E+01		
8	0.1873E+03	0.6759E+01	0.2057E+02	0.4700E+01		

COLD BOX HEAT BALANCE = -3.3223 Watt tsub= 5.5135
WORK DONE BY: T1, T2, T3 Watt = 54728.7281 25866.3529 11673.6043
WORK DONE BY WE Watt = 1900.6436

First, the simulation model has been checked upon several sets of experimentally known process conditions for 2-compressors and 3-compressors flows. The model has proven to be accurate within 5% when allowed to converge with sufficient number of iterations (normally greater than 7500). Then multiple runs have been performed to evaluate the coldbox liquid helium production to the Ring at different inlet conditions. The limiting factors of the present coldbox are: inlet pressure is less than 1.241 MPa, inlet flow is less than 1700 g/s, midline pressure is less than 0.62 MPa, and exit pressure is ~0.331 MPa psia. These limiting factors are reducing the overall potential increase of production resulting from JT valve replacement with an expander. The following table summarizes the results of simulation.

Parameters:	*2-compressors flow*		*3-compressors flow*	
Coldbox inlet flow, kg/s	**1.143**		**1.601**	
Coldbox inlet pressure, MPa	1.1376 (165 psia)		1.1376 (165 psia	
Coldbox inlet temp, °K	80.0		80.0	
Ring pressure, MPa	0.331 (48 psia)		0.331 (48 psia)	
Coldbox return temp, °K	4.7		4.7	
JT and WE split, %	**100% JT**	**100%WE**	**100%JT**	**100%WE**
Turbine 1 flow, kg/s	0.86265	0.89578	1.18907	1.24222
Turbine 2 flow, kg/s	0.43132	0.44789	0.59454	0.62111
Turbine 3 flow, kg/s	0.43132	0.44789	0.59454	0.62111
JT flow, kg/s	0.28035	0.00000	0.41192	0.00000
Expander flow, kg/s	0.00000	0.24722	0.00000	0.35878
Expander efficiency, %	75.0	75.0	75.0	75.0
Expander work, Watt	0.00000	1295.4873	0.00000	1832.7077
Turbine 1 speed, krpm	43095	44762	51993	53339
Turbine 2 speed, krpm	41785	43504	49849	51500
Turbine 3 speed, krpm	34861	35941	40933	41747
Turbine 1 efficiency, %	0.6392	0.6700	79.16	0.7991
Turbine 2 efficiency, %	0.6088	0.6381	75.25	0.7719
Turbine 3 efficiency, %	0.6481	0.6738	77.20	0.7790
Turbine 1 temp in, °K	43.83	44.79	43.68	44.50
Turbine 1 temp out, °K	36.12	36.78	35.43	36.09
Turbine 2 temp in, °K	24.53	25.19	24.49	25.02
Turbine 2 temp out, °K	16.20	16.51	15.81	16.13
Turbine 3 temp in, °K	12.58	12.66	12.58	12.59
Turbine 3 temp out, °K	8.00	8.0	8.00	8.00
Turbine 1 work, Watt	35376	38156	52142	55546
Turbine 2 work, Watt	18325	19902	26419	28310
Turbine 3 work, Watt	8583	9125	12039	12620
Flow to JT and WE, kg/s	0.28035	0.24722	0.41192	0.35878
Flow to the Ring, kg/s	**0.13880**	**0.15384**	**0.20291**	**0.22179**
Increase of production, %	**0.0**	**10.83**	**0.0**	**9.3**

It must be understood that the potential increase of production due to JT valve replacement with an expansion engine is a relative number, which is dependent upon the base conditions. The production increase due to JT replacement with a wet engine at present conditions and for 3-compressors flow is ~9.3%. Nevertheless, the calculations show that if the coldbox production is routed into a low pressure dewar at 33 psia, and wet expander efficiency is 65%, then the increase of production (WE vs. JT) is 10.8 %. The coldbox inlet pressure is another contributor, i.e. higher the inlet pressure, greater the expected increase of production. The calculations show that if CHL coldbox

is operated at 1.310 MPa inlet pressure (instead of present 1.1376 MPa) then the expected increase of production (WE vs. JT) would constitute another 1.5%. But the single greatest contributor to the effect of JT replacement with expander is a coldbox performance at initial conditions, e.g. how close the coldbox thermodynamic efficiency to the Carnot efficiency. If a coldbox is designed and operated far below its optimum for a given set of inlet/outlet conditions, then such a replacement would produce a greater relative increase of production than for an optimized coldbox. Therefore the increase of production as much as 28% reported in [5] should be understood in this context.

SUMMARY

A mathematical model based on a finite element approach has been developed to simulate the thermo- and gas-dynamic processes for the equipment included in a helium coldbox. The model provides for high accuracy and conversion stability. Though the coefficients used in thermo- and gas-dynamic equations are unique for the given coldbox, the general approach, the equations, the methods of computations, and most of the subroutines written in FORTRAN can be readily applied to different coldboxes. The simulation results are compared against actual operating data to demonstrate applicability of the model.

REFERENCES

1. C.H.Rode, Tevatron cryogenic system, "The Proceedings of the 12th International Conference on High Energy Accelerators" (1983).
2. G.A.Hodge, et al, Recent operating experience with the Fermilab Central Helium Liquefier, "Advances in Cryogenic Engineering" Vol.29, (1984).
3. R.J.Walker, et al, Recent operating experience with the Fermilab Central Helium Liquefier, "Advances in Cryogenic Engineering" Vol.31, (1986).
4. J.N.Makara, et al, Fermilab Central Helium Liquefier Upgrade, "Advances in Cryogenic Engineering" Vol.39A, (1994).
5. Venkatarao Ganni, et al, Capacity Upgrade of the Exell Helium Liquefier Plant by the Addition of a Wet Engine, "Advances in Cryogenic Engineering" Vol.31, (1985).